nanguTV

WORKSHOP

Multi-Platform React

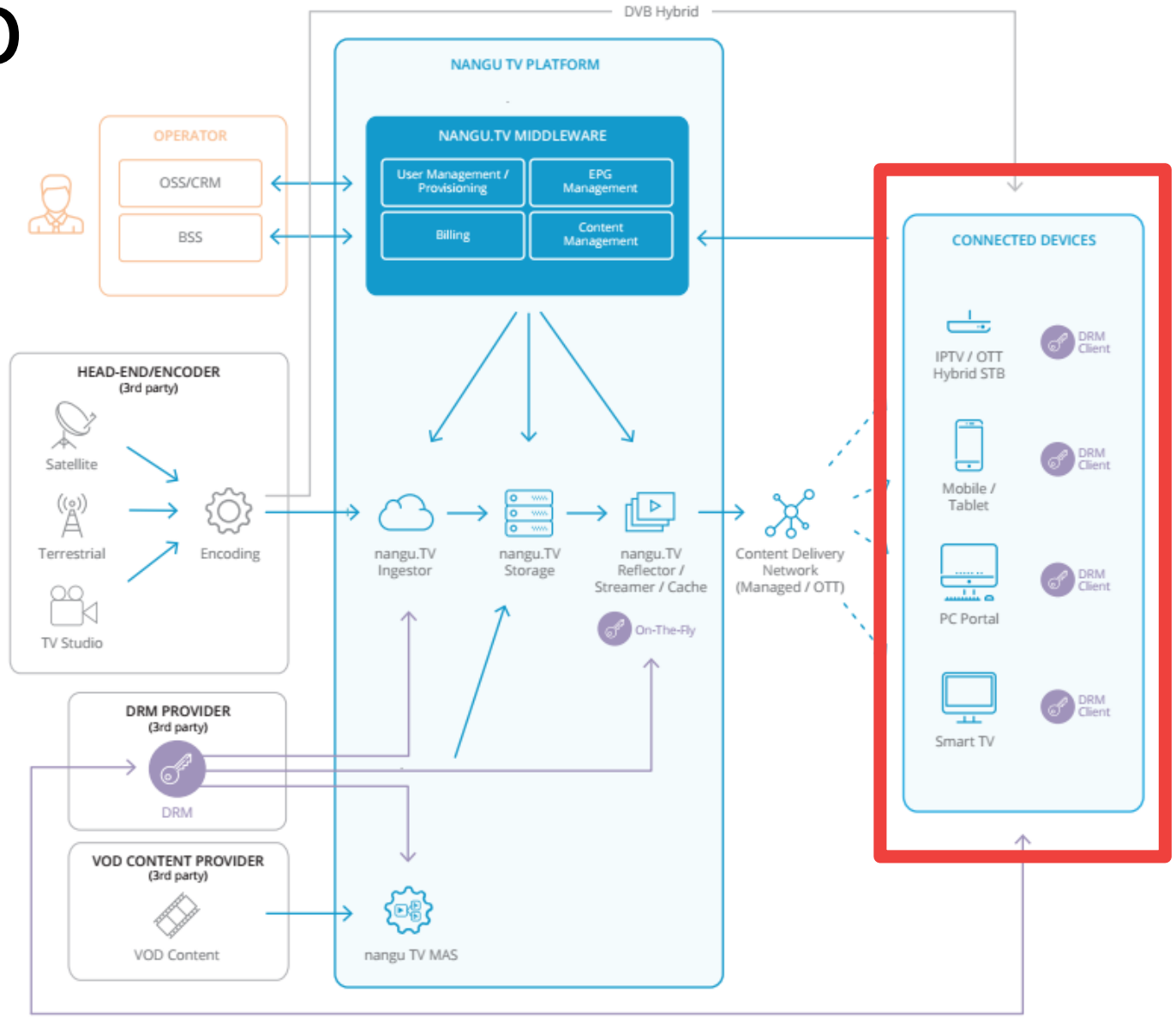
2019/11/1

Who are we?

- **Jakub Římal**
  - frontend team leader (web-based)
  - web, Smart TVs, set-top boxes

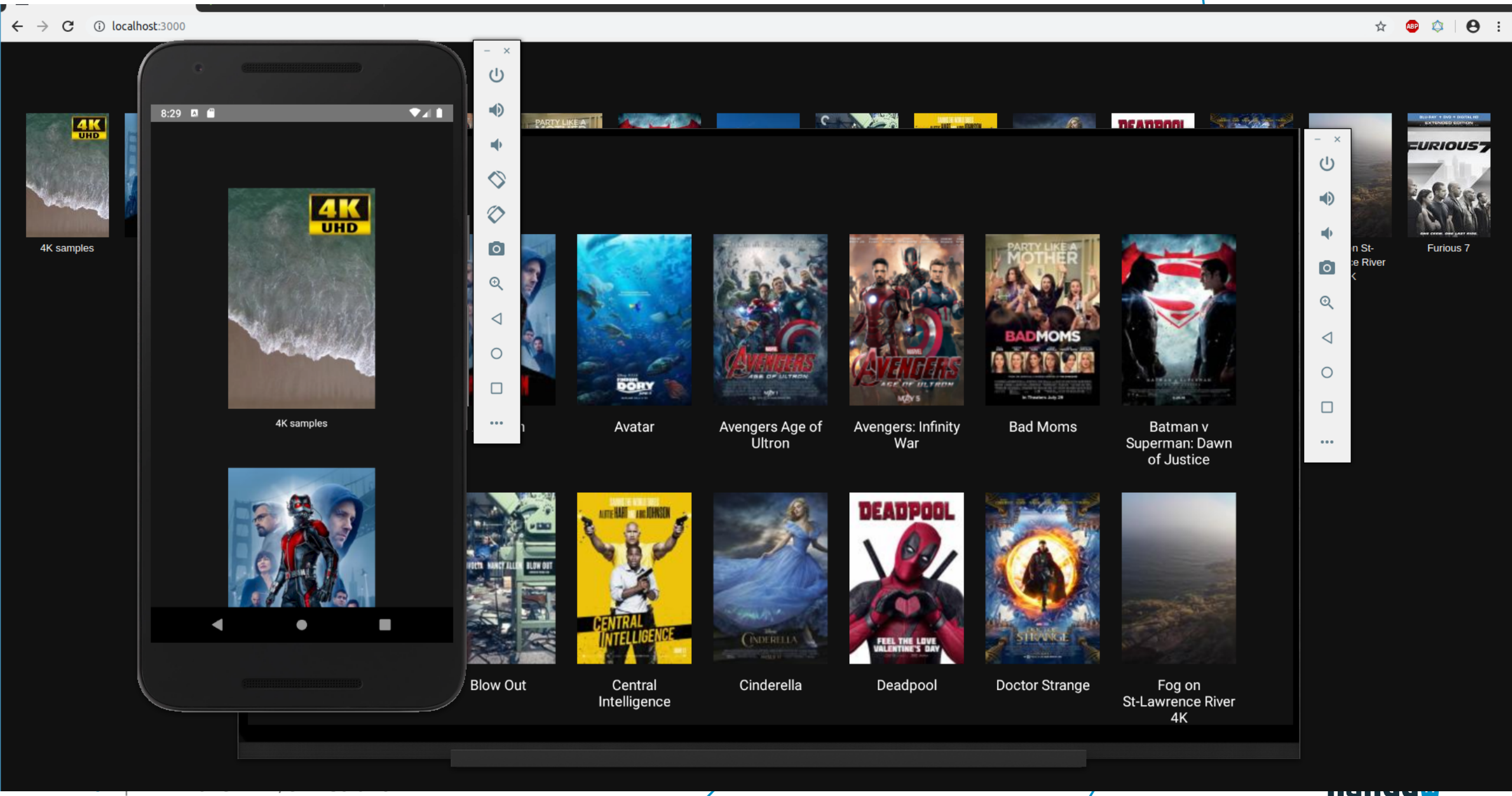
- **Matej Čepil**
  - frontend team leader (mobile & native)
  - Android, iOS, Apple TV, Android TV

What do we do?


DVB Hybrid
NANGU TV PLATFORM
OPERATOR
OSS/CRM
BSS
NANGU.TV MIDDLEWARE
User Management / Provisioning
EPG Management
Billing
Content Management
CONNECTED DEVICES
IPTV / OTT Hybrid STB
DRM Client
Mobile / Tablet
PC Portal
Smart TV
HEAD-END/ENCODER
(3rd party)
Satellite
Terrestrial
TV Studio
Encoding
nangu.TV Ingestor
nangu.TV Storage
nangu.TV Reflector / Streamer / Cache
On-The-Fly
Content Delivery Network (Managed / OTT)
DRM PROVIDER
(3rd party)
DRM
VOD CONTENT PROVIDER
(3rd party)
VOD Content
nangu TV MAS

Our plan for today

Create multi-platform app which runs on web, mobile and TV


localhost:3000
8:29
4K samples
4K samples
Avatar
Avengers Age of Ultron
Avengers: Infinity War
Bad Moms
Batman v Superman: Dawn of Justice
Blow Out
Central Intelligence
Cinderella
Deadpool
Doctor Strange
Fog on St-Lawrence River 4K
Furious 7

Our plan for today

1. Create simple React project
2. Add React Native to an existing React project
3. Implement a bit more complicated (add styles, more components) version of the app for web only
4. Add primitives
5. Use existing styles for native
6. Finalize styles and behavior so it looks and behave the same on both platforms
7. Create mobile layout and use it for both (native mobile app and web app running on mobile device)
8. (Optional) Add some animations

1. Create simple React project

```
yarn create react-app my-reactive-workshop
cd my-reactive-workshop
yarn && yarn start
```

Create simple React Native project

```
npx react-native init myNativeReactiveWorkshop
cd myNativeReactiveWorkshop
yarn start #keep_this_terminal_running
```

Now run your emulator (e.g. with Android Studio)

```
npx react-native run-android
```

2. Add React Native to an existing React project

(see http://facebook.github.io/react-native/docs/integration-with-existing-apps.html)

1. add script `"start-native": "react-native start"` to your package.json file
2. copy „android“ folder from the project created with react-native init and paste it to the project root
3. copy „app.json“ file from the project created with react-native init and paste it to the project root
4. merge „.gitignore“ file from the project created with react-native init with the one in the current project (probably copy everything except node.js section)
5. (optional) unify app/package name

2. Add React Native to an existing React project (2)

7. copy „App.js“ file from the project created with react-native init and paste it to the project root
8. copy „index.js“ file from the project created with react-native init, paste it to the project root and rename it to „index.native.js“
9. rename src/index.js file to src/index.web.js
10. rename all files containing web components from *.js to *.web.js
11. `yarn add react-native`

Now you should be able to start both React (with `yarn start`) and React native (with `yarn start-native`) apps but in fact it is just two different apps inside one project, see the rest of the workshop to have the real multi-platfrom app. :)

Clone the repository

In your workspace (next to your already existing projects):

```
git clone https://github.com/nangutv/reactive-workshop.git
cd reactive-worshop
git checkout step-1-basic-react-app
yarn && yarn start
```

Explore the project and ...

```
git checkout step-2-basic-react-native-app
yarn && yarn start-native #keep_this_terminal_running
```

Now run your emulator (e.g. with Android Studio)

```
npx react-native run-android
```

3. Implement a bit more complicated (add styles, more components) version of the app **for web only**

```
git checkout step-3-complete-web-app
```

4. Add primitives
5. Use existing styles for native

```
git checkout step-4-native-app-is-able-to-run
```

6. Finalize styles and behavior so it looks and behave the same on both platforms

```
git checkout step-5-native-app-works-completely
git checkout step-6-correct-styling-everywhere
```

7. Create mobile layout and use it for both (native mobile app and web app running on mobile device)

```
git checkout step-7-special-style-for-mobile
```

8. (Optional) Add some animations

And what about adding React to an existing React Native project?

See the „start-from-native“ git branch for more:

```
git checkout start-from-native
```

Summary

1. Abstraction (for component)
2. Abstraction (for styles)
3. Abstraction (for everything where you need something platform-specific)

Thank you